PROJECT REPORT

Name: Syed Muhammad Raza

Student Id: 63558

Name: Syed Muhammad Ghazi

Student Id: 9735

Subject: Artificial Intelligence

Project Name: Helmet Detection

Helmet Detection

Introduction:-

In Helmet Detection, we will train our custom dataset for the detection. The motive is to detect the people who wearing helmets on main roads and police will easily find those who don't wear a helmet and will take strict action against them. We will use Yolov3 model of training purpose and our dataset consist of more than 500 images.

We made the videos of a helmet and, by using Python script we break the video into images. In the second step, annotation of the dataset by using the Label Image Tool after labeling the dataset, then a process of training will start using a powerful GPU (GTX 1660 Super X).

Tools:-

1. **Python**
   By using python we write two programs for our project, one for real-time helmet detection and another for the helmet detection using video.

2. **Yolo**
   We trained our data set on Yolo architecture with the best's result of mean average precision and benchmark.

3. **Ubuntu**
   We use Ubuntu operating system for the project. Because it professional and easy to train the data set and run on it.

4. **Image label tool (Annotation)**
   Image label tool is used to annotate the data set correctly in order to train and test the data set.

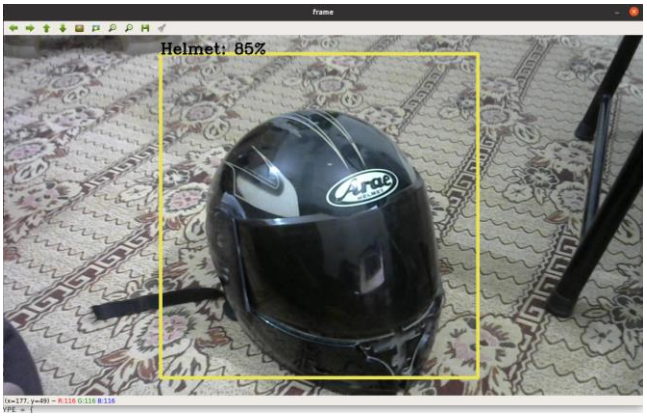
5. **Darkflow**

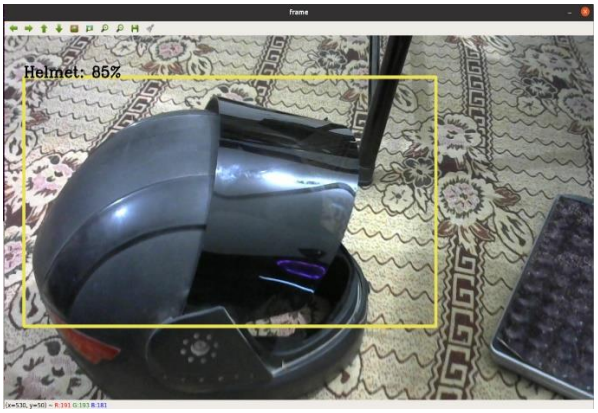
   Darkflow is used to for the real time helmet detection because the Yolo few version do not support the real time helmet detection. We use this as a secondary.

Problem:-

Nowadays, many accidents are being reported. It has been reported that the majority of cases are youngsters because they do not take safety precautions and not wear the helmet with riding on a motorcycle. According to a survey, studies have shown that wearing a bike helmet reduces your risk of serious head injury by about 70 percent, to avoid motorcycle accidents we are trying to solve this problem by using Helmet detection.

Results:-





Code:

Real Time Helmet Detection:-

```
from darkflow.net.build import TFNet
import matplotlib.pyplot as plt
import tf_slim as slim
import time
import numpy as np
import os
import cv2

options = {
    'model': 'yolov2-obj-helemt.cfg',
    'load':'bin/yolo-obj_best-helmet.weights',
    'threshold': 0.8,
    'gpu': 0
}

tfnet = TFNet(options)

filename = 'video125.avi'
frames_per_second = 244.0
res = '720p'

# Set resolution for the video capture
# Function adapted from https://kirr.co/0l6qmh
def change_res(capture, width, height):
    capture.set(3, width)
    capture.set(4, height)

# Standard Video Dimensions Sizes
```

```
STD_DIMENSIONS =  {
    "480p": (640, 480),
    "720p": (1280, 720),
    "1080p": (1920, 1080),
    "4k": (3840, 2160),
}


# grab resolution dimensions and set video capture to it.
def get_dims(capture, res='1080p'):
    width, height = STD_DIMENSIONS["480p"]
    if res in STD_DIMENSIONS:
        width,height = STD_DIMENSIONS[res]
    ## change the current caputre device
    ## to the resulting resolution
    change_res(capture, width, height)
    return width, height

# Video Encoding, might require additional installs
# Types of Codes: http://www.fourcc.org/codecs.php
VIDEO_TYPE = {
    'avi': cv2.VideoWriter_fourcc(*'XVID'),
    #'mp4': cv2.VideoWriter_fourcc(*'H264'),
    'mp4': cv2.VideoWriter_fourcc(*'XVID'),
}

def get_video_type(filename):
    filename, ext = os.path.splitext(filename)
    if ext in VIDEO_TYPE:
      return  VIDEO_TYPE[ext]
    return VIDEO_TYPE['avi']



capture = cv2.VideoCapture(0)
path = 'Helmet-DetectingImage/'
output = cv2.VideoWriter(filename, get_video_type(filename), 25, get_dims(capture
, res))

colors = [tuple(255 * np.random.rand(3)) for _ in range(5)]

i = 0
frame_rate_divider = 3
while(capture.isOpened()) :
    stime = time.time()
```

```
    ret, frame = capture.read()
    if ret:
        if i % frame_rate_divider == 0:
            results = tfnet.return_predict(frame)

            for color, result in zip(colors, results):
                tl = (result['topleft']['x'], result['topleft']['y'])
                br = (result['bottomright']['x'], result['bottomright']['y'])
                label = result['label']
                confidence = result['confidence']
                text = '{}: {:.0f}%'.format(label, confidence * 100)
                frame = cv2.rectangle(frame, tl, br, color, 5)
                frame = cv2.putText(frame, text, tl,cv2.FONT_HERSHEY_COMPLEX, 1,
(0, 0, 0), 2)
                image = np.array(frame)
            if confidence > 0.9 and label == "Helmet" :
               cv2.imwrite(os.path.join(path , 'helmet.png'), image)
            output.write(frame)
            cv2.imshow('frame', frame)
            print('FPS {:.1f}'.format(1 / (time.time() - stime)))
            i +=1
        else:
            i +=1
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
    else:
        break



#while True:
#    ret, frame = cap.read()
#    out.write(frame)
#    cv2.imshow('frame',frame)
#    if cv2.waitKey(1) & 0xFF == ord('q'):
#        break


capture.release()
output.release()
cv2.destroyAllWindows()
```

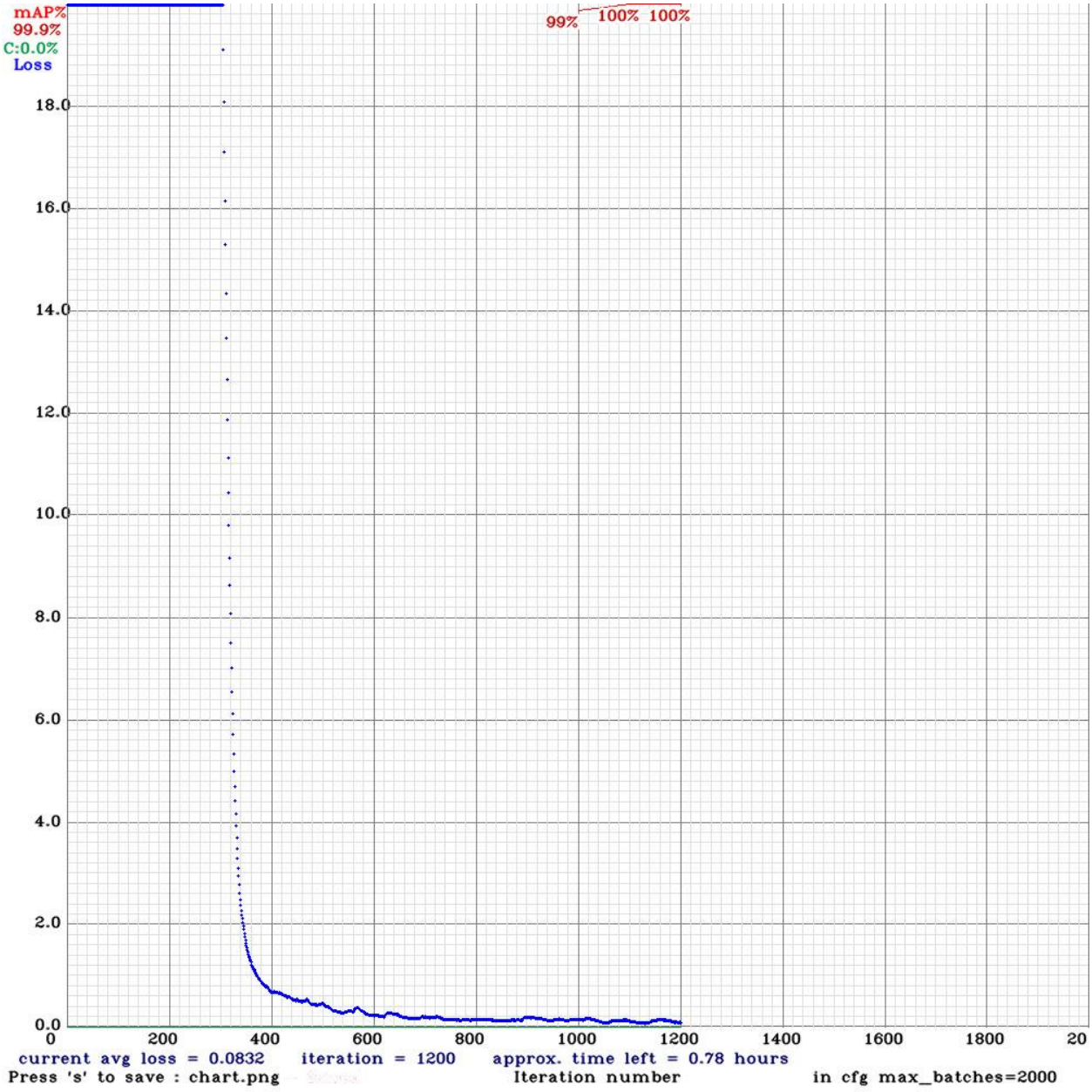
Helmet Detection using Video:-

```
import cv2
from darkflow.net.build import TFNet
import matplotlib.pyplot as plt
import numpy as np
import time
import os
import tf_slim as slim

options = {
 'model': 'yolov2-obj-helemt.cfg',
 'load': 'bin/yolo-obj_best-helmet.weights',
 'threshold': 0.6,
 'gpu': 0

}
tfnet = TFNet(options)
cap = cv2.VideoCapture('h1.mp4')
path = 'HelmetDataset/'
colors=[tuple(255 * np.random.rand(3)) for i in range(5)]
while(cap.isOpened()):
    stime= time.time()
    ret, frame = cap.read()
    results = tfnet.return_predict(frame)
    if ret:
        for color, result in zip(colors, results):
            tl = (result['topleft']['x'], result['topleft']['y'])
            br = (result['bottomright']['x'], result['bottomright']['y'])
            label = result['label']
            confidence = result['confidence']
            frame= cv2.rectangle(frame, tl, br, color, 7)
            frame= cv2.putText(frame, label, tl, cv2.FONT_HERSHEY_TRIPLEX, 1, (0,
0,0), 2)
        cv2.imshow('frame', frame)
        image = np.array(frame)
        if confidence > 0.8 and label == "Helmet" :
          cv2.imwrite(os.path.join(path , 'helmet.png'), image)
        print('FPS {:1f}'.format(1/(time.time() -stime)))
        if cv2.waitKey(1)  & 0xFF == ord('q'):
            break
    else:
        break
cap.release()
cv2.destroyAllWindows()
```

Graph:-



Conclusion:-

Helmet detection, mostly in every developed country helmet detection is implemented to charge fine who are not wearing helmet during riding a bike. This is the main motive behind our project is safety of the people who do not wear helmet while riding.